

11601
11-62-11
037717

Report for Neural Network Prototyping Package within IRAF

Final Report
Contract Number NAS5-32067David Bazell, Principal Investigator
General Sciences Corporation
July 1, 1997

Purpose of the Contract:

The purpose of this contract was to develop a neural network package within the IRAF environment to allow users to easily understand and use different neural network algorithms the analysis of astronomical data. The package was developed for use within IRAF to allow portability to different computing environments and to provide a familiar and easy to use interface with the routines. In addition to developing the software and supporting documentation, we planned to use the system for the analysis of several sample problems to prove its viability and usefulness.

Summary of Activities:

The initial development of the software prototype was done in the IRAF programming language SPP. We used SPP to develop an interface to several existing public domain neural network packages. This involved performing some unusual programming tasks such as redirecting the output of the existing programs using SPP system calls. This procedure turned out to be rather cumbersome, difficult to explain in documentation, and difficult to maintain. Nevertheless, the prototype code that was developed in this manner worked and was used to classify some sample IRAS LRS spectra.

Following this development phase we communicated several times with the IRAF help desk to determine a better approach. We were encouraged to write some IRAF scripts to interface to the existing neural network programs that were written in C. This turned out to be the better of the two approaches and is the one that was used in the final software that was delivered. The script approach was easier to use, maintain, and understand. Some changes were made to the neural network C routines to increase the compatibility with the IRAF environment. In particular, several interactive calls that were in the C routines were removed so that all necessary parameters could be specified (e.g., using the IRAF task eparam) when the program was initially run. Also, some monolithic C routines were broken up into a more modular set of routines and a script was used to run each one individually.

The three neural networks that we have investigated are the back-propagation network, the most commonly used type of neural network; learning vector quantization, and self organizing map. The back-propagation code was obtained from the University of Nevada at Reno, while the LVQ and SOM codes were obtained from Teuvo Kohonen's laboratory in Finland. LVQ and SOM are supervised and unsupervised versions of the Kohonen topology conserving map. The IRAF tasks that we implemented include a single task for running the back-propagation code, and several tasks for LVQ and SOM. The LVQ package has tasks called init to initialize the codebook vectors or

weights, balance to balance codebook vectors, lvqtrain to train the network with a set of examples, accuracy to test the accuracy of the training, and classify to classify a set of unknown examples using the previously trained codebook vectors.

In addition to the development of the IRAF scripts and modification of the C code, all the resulting IRAF tasks were documented using IRAF help files. Each help file contains the standard description of the routine: a detailed explanation of how each routine works, discussion of each of the input and output parameters, and examples of how the routine can be run under various circumstances. We also provided a detailed installation document, giving a step by step explanation of how the software can be installed. Of course, familiarity with IRAF installation procedures will be helpful.

Scientific Problems Explored:

We developed approaches to several problems of scientific interest using of neural network package. The first example we explored was classification of IRAS LRS spectra using back-propagation. The samples of LRS spectra we had fell into 5 different classes of rather different looking spectra. We found that back-propagation had no trouble classifying these spectra into their known categories. Further work could be done on this project by selecting spectra that were more similar in appearance, thus making it harder for the algorithm to differentiate them. We plan to explore this in the future.

We also attempted to perform a deconvolution problem by training a back-propagation network to produce a narrow gaussian in response to being trained with spectra of point sources that were broadened by a point spread function. In this way the network learns the mapping from broadened point source image to point source, effectively learning to deconvolve the image in the process. The hope would be that this internal mapping would then be applicable to spectra of non-point sources. We were moderately successful in this endeavor, being able to deconvolve broad band spectra which compared reasonable well with other methods of deconvolution. We plan to investigate this further with an aim toward publication.

Our main project was the classification of stars and galaxies using both back-propagation and LVQ. Here we used 87 objects consisting of 60 galaxies and 27 stars. All samples have a similar brightness level. All objects were checked against a database of stars and galaxies to confirm the type of object. The coordinates of each object were recorded in J2000.

In the star-galaxy classification project we investigated several different pre-processing methods. These took the raw data in image format and extracted information that was then input into the different neural network models for classification. The pre-processing methods are as follows:

- a) The first method involves finding the brightest pixel in the object and producing a linear profile of the objects by plotting pixel brightness as a function of distance from the brightest pixel. Here the distance is calculated by looking at concentric boxes around the brightest pixel.
- b) The second method is similar to 2) beginning with plotting the brightness as a function of distance from the brightest pixel. We then calculate the distance between the brightest pixel and every other pixel in the image, which produces a discrete set of distances due to the pixelization. Each discrete distance bin is averaged to produce a single value and this value is plotted as a

function of distance. This produces a profile of object brightness which is used as input to the networks.

c) Again the pixels are binned as in b), but the averaged brightness for each bin is subtracted from all the values in the bin. If a given pixel has above average brightness it is given a positive value, if below average it is given a negative value. The brightnesses are then summed to give one value for each bin. These brightness differences are then used as input to the networks.

d) Bin as in b), then find the derivative of the intensity and use this as input to the network.

These four pre-processing methods were compared with each other and with the a using the raw data as input to the networks. We compared back-propagation and learning vector quantization networks using these data. Detailed discussion of our methods and results will be appearing in a paper in the *Astrophysical Journal*.

Disposition of the Software

The software developed under this contract has been delivered to the IRAF ftp site at NOAO. It is freely available for anyone who wants to use it.